Universidad de Costa Rica

Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE-0624 –Laboratorio de Microcontroladores
II ciclo 2023

Laboratorio 4

STM32: GPIO, ADC, comunicaciones, Iot

Sofía Villalta Jinesta B88565
Elias Alvarado Vargas B80372
Profesor: Marco Villalta Fallas

29 de Octubre de 2023

Índice

Índice de figuras

Índice de tablas

1. Resumen

En este tercer laboratorio se busca desarrollar un sismógrafo digital para registrar y estudiar las oscilaciones en el edificio de la Escuela de Ingeniería Eléctrica, entonces se utiliza una placa STM32F429 Discovery kit y la biblioteca libopencm3, además de un botón para habilitar o deshabilitar comunicaciones por USART/USB, seguidamente un led que debe parpadear indicando la habilitación de la transmisión y recepción de datos por el periférico USART/USB. Como parte del funcionamiento es capaz de leer los ejes del giroscopio (X, Y, Z), además de leer el nivel de la batería cuyo rango es de [0,9]V, en el caso de estar cerca del límite mínimo de operación del microcontrolador de 7 V debe encender un led de alarma parpadeante y enviar la notificación de batería baja al dashboard de thingsborad, por lo que se despliega en la pantalla LCD el nivel de batería, los valores de los ejes X, Y, Z y por último se crea un script de python que envía la información del giroscopio y nivel de batería para ser desplegados en la plataforma Iot thingsboard.

2. Nota Teórica

En la presente sección, se muestra el microcontrolador utilizado, además de los componentes y periféricos usados, con el diseño del circuito final.

2.1. Microcontrolador STM32F427

El STM32F427 es un tipo de microcontrolador y es desarrollado por STMElectronics, además cuenta con un núcleo Arm Cortex M4 CPU de 32 bits, tiene una memoria flash de 2 MB, funciona a una frecuencia de 180 MHz [1]. La alimentación tiene un rango de valores entre 1.7 V a los 3.6 V, posee también una memoria SRAM de 254 KB y un controlador de memoria externa flexible con memoria de datos de hasta 32 bits, el digrama de pines que conforma este microcontrolador se muestra en la siguiente Figura 3

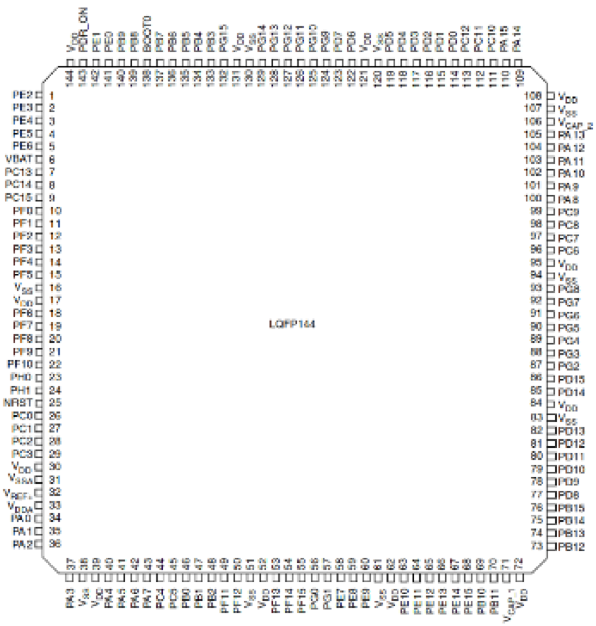


Figura 1: Pines del microcontrolador. Recuperado de [1]

El diagrama de bloques es el siguiente

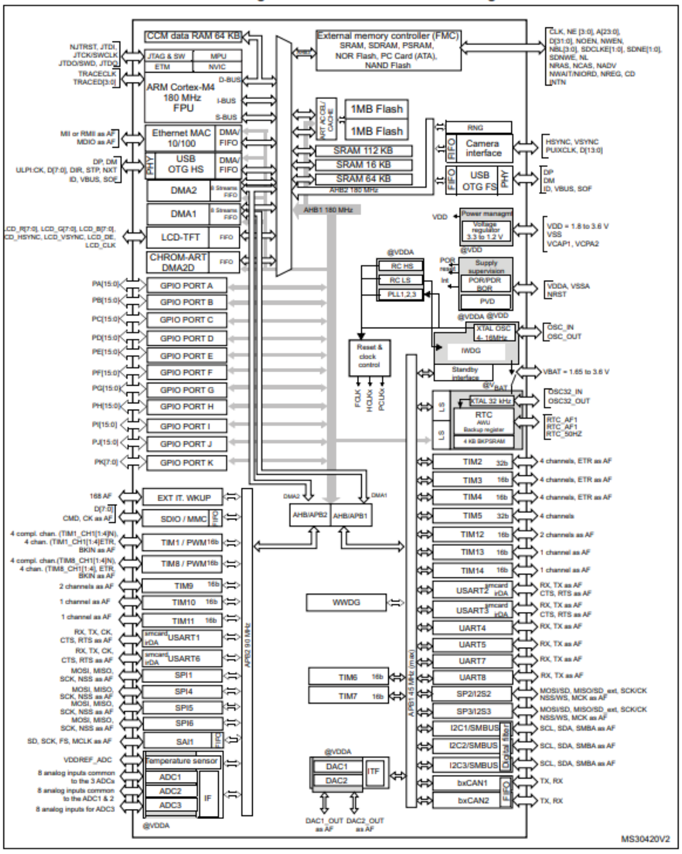


Figura 2: Diagrama de bloques. Recuperado de [1]

Continuando con la placa STM32F429 Discovery Kit, lo utiliza el MCU STM32 con el objetivo de brindar un tipo de dispositivo que logre desarrollar aplicaciones con una gran cantidad de periféricos[2]. La placa es la siguiente:

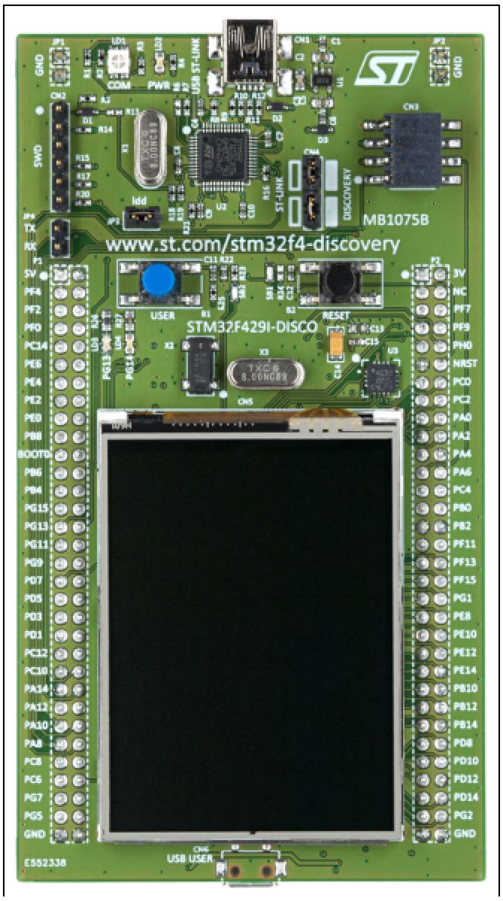


Figura 3: Placa STM32F429 Discovery Kit. Recuperado de [2]

Algunos periféricos que vienen incorporados a la tarjeta son los siguientes:

- Dos botones
- SDRAM de 64 Mbits
- 6 LEDs
- Encabezado de extensión para E/S LQFP144 para una conexión rápida a la placa de creación de prototipos
- MCU - STM32
- Con la placa una alimentación que puede tener la opción de ser una alimentación externa de 3V o 5V o a través de un bus USB.
- L3GD20, sensor de movimiento
- USB OTG con conector micro-AB

Giroscopio

El L3GD20 representa un sensor de velocidad angular de tres ejes que cuenta con un bajo consumo de energía, tiene un elemento de detección y una interfaz IC que es capaz de proporcionar la tasa angular que se mide a lo externo con la interfaz serial I2C/SPI [2]. Entonces el MCU STM32 se encarga de controlar el sensor mencionado anteriormente a través de la interfaz SPI.

Pantalla LCD

Este kit del microcontrolador incluye una pantalla LCD de tamaño 2.4 pulgadas con tecnología LCD TFT y con una resolución QVGA de 240x320 pixeles[2].

2.2. Lista de Componentes

Tabla 1: Tabla 1: Lista de Componentes

Componente	Tipo	Cantidad	Precio por unidad	Lugar
Potenciómetro	10 kΩ	1	₡350	Steren
Resistencia	10 kΩ	1	$0,05	Microjpm
Resitencia	18 kΩ	1	$0,05	Microjpm
Batería	9 V	1	₡2.300	Steren
Microcontrolador	STM32F429 Discovery Kit	1	₡42 515	Amazon

2.3. Otros componentes electrónicos adicionales

- Resistores: Es un elemento que produce una resistencia al paso de corriente eléctrica, por lo que se puede determinar el paso de corriente si se conoce el valor de la tensión y de la resistencia, aplicando ley de ohm [3].

2.4. Diseño del Circuito

En relación al diseño del circuito primero se menciona la sección de hardware diseñada, primeramente se considera como entrada valores de tensión que pueden variar entre 0 a 9 V dependiendo de la carga de la batería, por lo que se implementó un divisor de tensión para lograr bajar los niveles de tensión a los aceptables para el microcontrolador STM32429i [0,3.3] V, también se utilizó un potenciómetro para poder simular el comportamiento de la batería al descargarse, se utilizó el giroscopio para leer el ángulo de inclinación en x,y y z para después mostrarlo a través de la pantalla lcd junto con la lectura de tensión de la batería. Después se implementó la comunicación serial a través de la UART para transmitir los valores leídos a la plataforma de thingsboard.

2.4.1. Diseño de divisor de tensión

Para el diseño de el divisor de tensión se implementó la siguiente ecuación para tener un Vout de 3.3 V como máximo:

Dada la ecuación del divisor de tensión:

$$V_{out} = V_{in} \times \frac{R_2}{R_1 + R_2} \tag{1}$$

Si $V_{in} = 9V$, $V_{out} = 3{,}3V$, y $R_2 = 10k\Omega$, podemos sustituir estos valores en la ecuación:

$$3{,}3 = 9 \times \frac{10k}{R_1 + 10k} \tag{2}$$

Multiplicando ambos lados por $R_1 + 10k$:

$$3{,}3(R_1 + 10k) = 9 \times 10k \tag{3}$$

Expandiendo y simplificando:

$$3{,}3R_1 + 33k = 90k \tag{4}$$

Restando 33k de ambos lados:

$$3{,}3R_1 = 57k \tag{5}$$

Dividimos por 3.3 para despejar R_1:

$$R_1 \approx 17{,}27k\Omega \tag{6}$$

Por lo que el valor de R2 se decidó que fuera de 18k. El diseño planteado se puede observar en la figura 4 donde se puede observar que con la simulación se obtiene la tensión deseada.

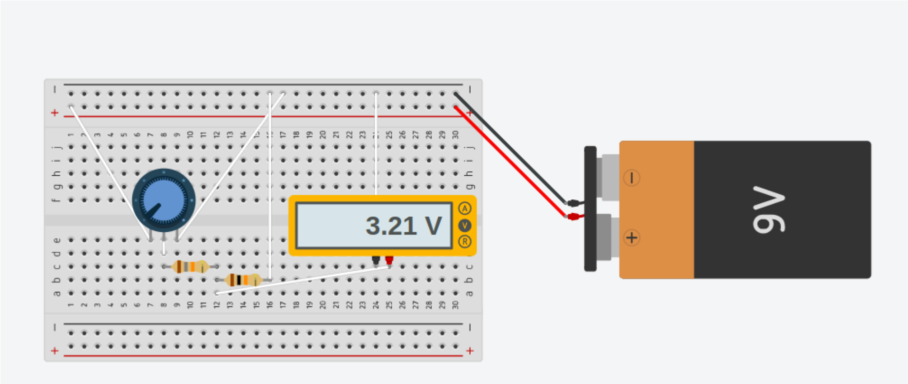


Figura 4: Diseño de circuito con potenciómetro y regulador de tensión

Ya con esta etapa diseñada se conectó el microcontrolador al circuito, como se puede observar en la figura 5:

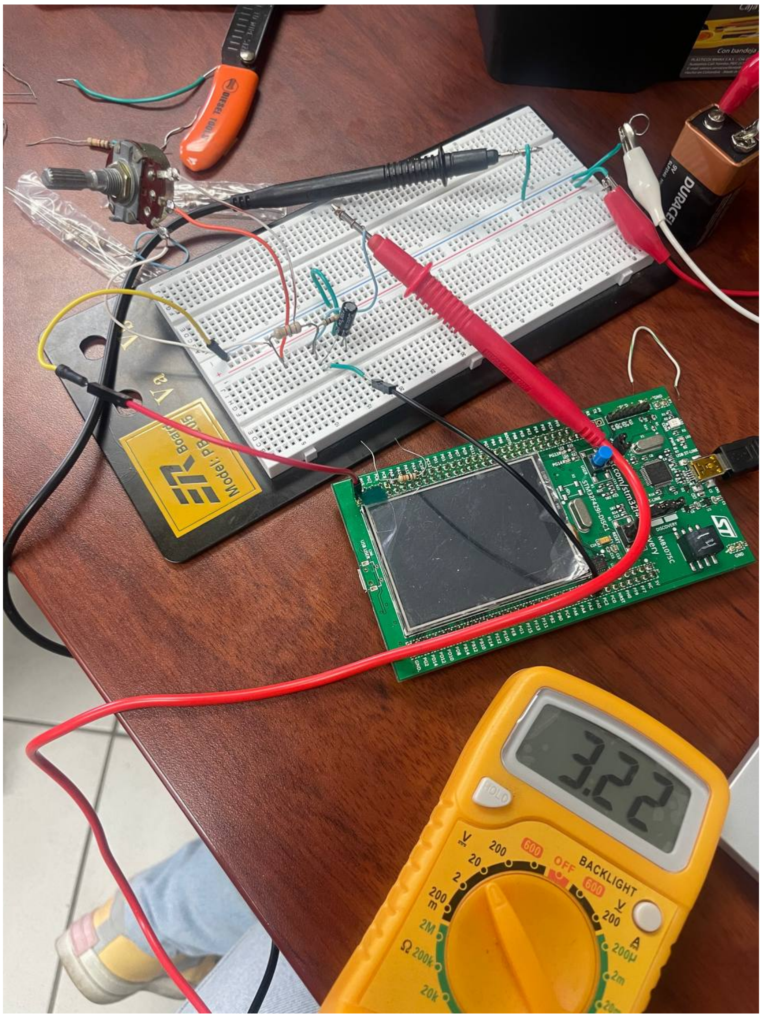


Figura 5: Circuito implementado con microcontrolador

2.5. Implementación del giroscopio

Para el manejo del giroscopio se utilizaron funciones del ejemplo gyr_spi. El código implementado se puede observar a continuación:

```
#include "gyr_spi.c"
#include "gyr_spi.h"
x = gyr_readX();
y = gyr_readY();
z = gyr_readZ();

sprintf(lcd_out, "X: %.2f", (float)x/100);
gfx_setCursor(15, 36);
gfx_puts(lcd_out);
sprintf(lcd_out, "Y: %.2f", (float)y/100);
gfx_setCursor(15, 90);
gfx_puts(lcd_out);
sprintf(lcd_out, "Z: %.2f", (float)z/100);
gfx_setCursor(15, 144);
gfx_puts(lcd_out);
```

Se decidieron manejar los valores en decimales,los resultados de esta implementación se pueden observar a la continuación:

Figura 6: Pantalla mostrando el giroscopio

2.6. Implementación de lectura de tensión de batería

Para la lectura de la batería se conectó la salida del divisor de tensión al puerto PA7 del microcontrolador, esto debido a que en la hoja de datos de establece que es un pin válido para usar el ADC se hace una conversión para poder mostrar en pantalla el valor original de la batería antes de pasar por el divisor de tensión. El código implementado se muestra a continuación:

```
#define VREF 3.3f
#define ADC_MAX_VALUE 4095.0f
#define CONVERSION_FACTOR 2.703f
#define NUM_SAMPLES 100
#define BATTERY_THRESHOLD 7.5f
#define LED_BLINK_INTERVAL 100000 //medio segundo

float battery_voltage;

static uint16_t read_adc_naiive(uint8_t channel) {
    uint8_t channel_array[16];
    channel_array[0] = channel;
    adc_set_regular_sequence(ADC1, 1, channel_array);
    adc_start_conversion_regular(ADC1);
    while (!adc_eoc(ADC1));
    return adc_read_regular(ADC1);
}

```

```c
uint16_t read_adc_average(uint8_t channel) {
    uint32_t sum = 0;
    for (int i = 0; i < NUM_SAMPLES; i++) {
        sum += read_adc_naiive(channel);
    }
    return sum / NUM_SAMPLES;
}

float adc_to_voltage(uint16_t adc_value) {
    float voltage = (adc_value / ADC_MAX_VALUE) * VREF;
    return voltage * CONVERSION_FACTOR;
}

void adc_setup(void) {
    rcc_periph_clock_enable(RCC_ADC1);
    gpio_mode_setup(GPIOA, GPIO_MODE_ANALOG, GPIO_PUPD_NONE, GPIO7);
    adc_power_off(ADC1);
    adc_enable_scan_mode(ADC1);
    adc_set_continuous_conversion_mode(ADC1);
    adc_disable_external_trigger_regular(ADC1);
    adc_set_right_aligned(ADC1);
    adc_set_sample_time_on_all_channels(ADC1, ADC_SMPR_SMP_28CYC);
    adc_power_on(ADC1);
}


```

Después de implementar esto en el código los resultados de esta implementación fueron los siguientes:

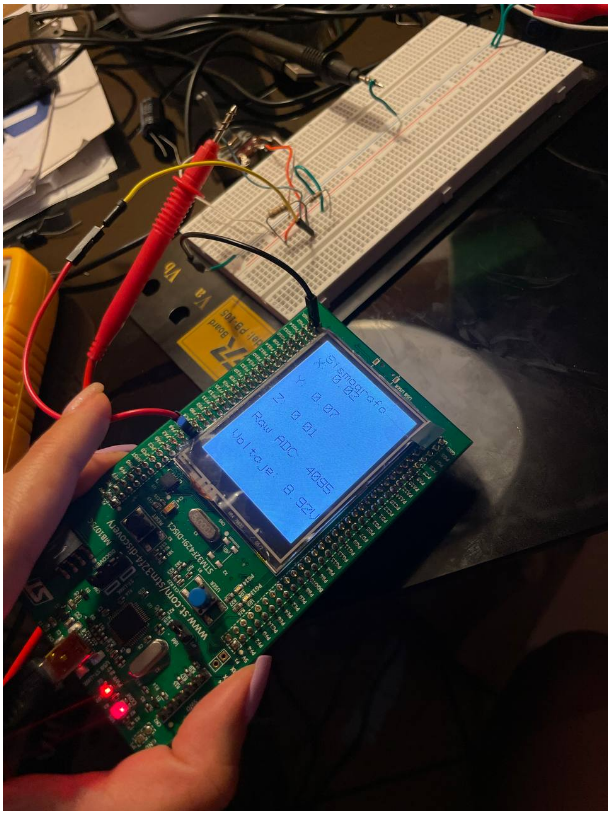


Figura 7: Pantalla mostrando el valor de la batería y sismógrafo

Para la parte del led de emergencia en el caso de batería mayor a 7v. Se configuró para que el led que parpadeara fuera el led rojo asociado al pin de PG14, como se puede observar a continuación:

```c
void gpio_setup_for_ledR(void) {
    rcc_periph_clock_enable(RCC_GPIOG);
    gpio_mode_setup(GPIOG, GPIO_MODE_OUTPUT, GPIO_PUPD_NONE, GPIO14);
}
```

En la función principal se implementó el siguiente código, que está basado en el ejemplo de miniblink:

```c
        if (battery_voltage < 7.0f) {
            gpio_toggle(GPIOG, GPIO14);  // Hace que el LED parpadee
            for (int i = 0; i < LED_BLINK_INTERVAL; i++) {
                __asm__("nop");
            }
        } else {
            gpio_clear(GPIOG, GPIO14);  // Apaga el LED
        }
```

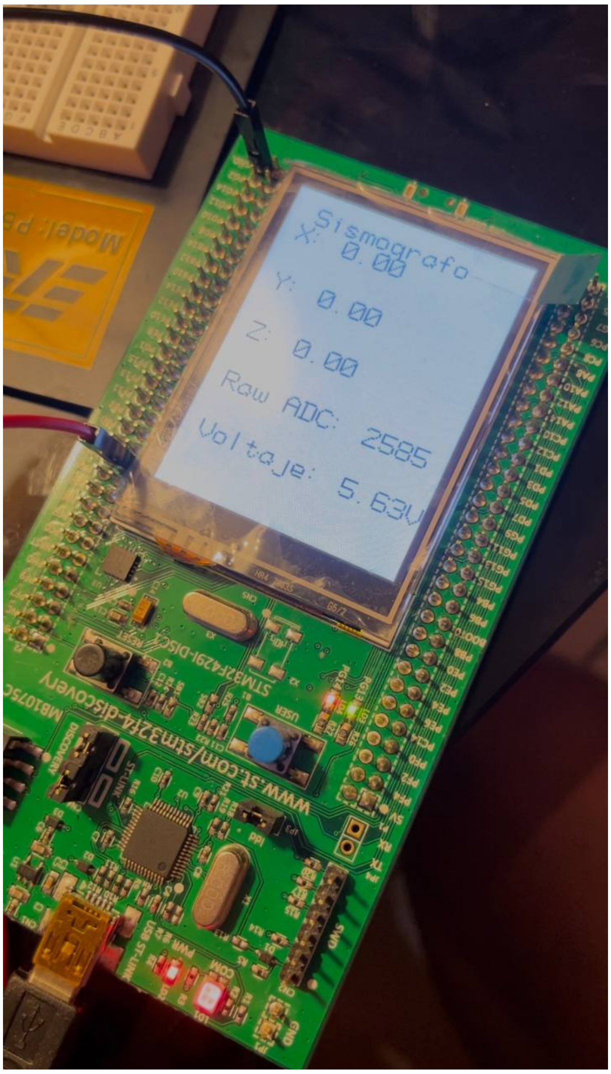


Figura 8: Led de alarma por batería baja

2.7. Implementación de la UART y thingsboard

Para esta parte se implementó la transmisión serial a través de la UART, utilizando el botón, que de acuerdo con la hoja del fabricante está conectado al pin PA0 por lo que se utilizó para la configuración mediante el siguiente código:

```c
static void usart_setup(void)
{
  /* Setup GPIO pins for USART1 transmit. */
  gpio_mode_setup(GPIOA, GPIO_MODE_AF, GPIO_PUPD_NONE, GPIO9);

  /* Setup USART1 TX pin as alternate function. */

  gpio_set_af(GPIOA, GPIO_AF7, GPIO9);

  usart_set_baudrate(USART1, 115200);
  usart_set_databits(USART1, 8);
  usart_set_stopbits(USART1, USART_STOPBITS_1);
  usart_set_mode(USART1, USART_MODE_TX);
  usart_set_parity(USART1, USART_PARITY_NONE);
  usart_set_flow_control(USART1, USART_FLOWCONTROL_NONE);
  /* Finally enable the USART. */
  usart_enable(USART1);
    gpio_set(GPIOG, GPIO13);
}

void usart_send_string(char* str)
{
    while(*str)
    {
        usart_send_blocking(USART1, *str++);
        gpio_toggle(GPIOG, GPIO13);  // Hace que el LED parpadee
            for (int i = 0; i < LED_BLINK_INTERVAL; i++) {
                __asm__("nop");
        }
    }
}

void button_setup(void) {
    rcc_periph_clock_enable(RCC_GPIOA);
    gpio_mode_setup(GPIOA, GPIO_MODE_INPUT, GPIO_PUPD_NONE, GPIO0);
}
bool is_button_pressed(void) {
    return gpio_get(GPIOA, GPIO0);
    }
```

3. Análisis de Resultados

En la presente sección de analiza de forma detallada lo que se desarrolló en el laboratorio 4.

3.1. Desarrollo del programa

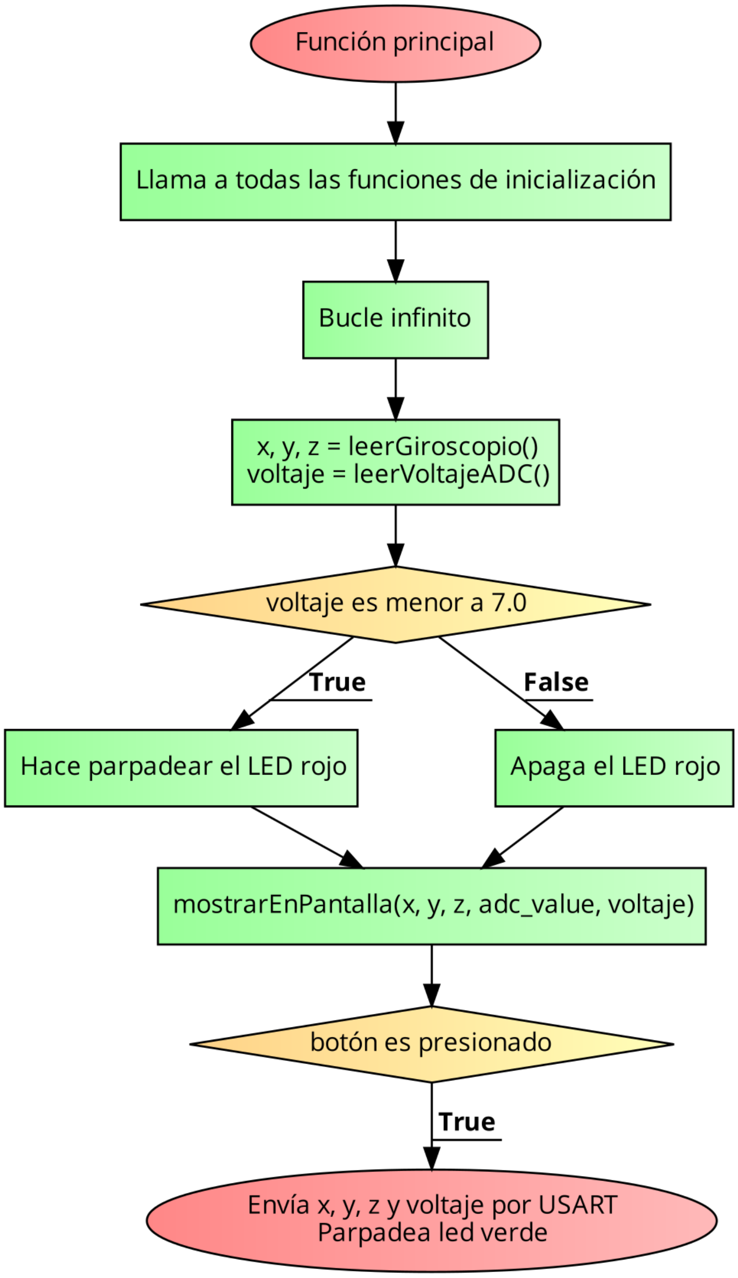


Figura 9: Diagrama del código implementado

Como se puede observar en la figura 11 el programa principal comienza definiendo algunos valores constantes y inicializando las funciones de configuración, después dentro de un bucle infinito mientras el dispositivo está conectado se leen los valores del giroscopio, se guardan en las variables x, y y z que se imprime en la pantalla LCD mediante el uso de sprintf, también se encarga de leer el valor de la tensión y encender el led de alarma rojo si es menor a 7v, cuando el botón es presionado comienza la transmisión serial. Como se oberva en el siguiente código:

```
int main(void) {
    char lcd_out[100];
    char usart_out[200];
    int16_t x, y, z;

    clock_setup();
    console_setup(115200);
    sdram_init();
    gyr_setup();
    adc_setup();
    lcd_spi_init();
    gfx_init(lcd_draw_pixel, 240, 320);
    gpio_setup_for_ledR();
    gpio_setup_for_ledG();
    usart_setup();

    while (1) {
        gfx_fillScreen(LCD_WHITE);
        gfx_setTextSize(2);
        gfx_setCursor(25, 15);
        gfx_puts("Sismografo");

        x = gyr_readX();
        y = gyr_readY();
        z = gyr_readZ();

        sprintf(lcd_out, "X: %.2f", (float)x/100);
        gfx_setCursor(15, 36);
        gfx_puts(lcd_out);
        sprintf(lcd_out, "Y: %.2f", (float)y/100);
        gfx_setCursor(15, 90);
        gfx_puts(lcd_out);
        sprintf(lcd_out, "Z: %.2f", (float)z/100);
        gfx_setCursor(15, 144);
        gfx_puts(lcd_out);

        uint16_t raw_adc_value = read_adc_average(7); // Canal 7
        battery_voltage = adc_to_voltage(raw_adc_value);

        if (battery_voltage < 7.0f) {
            gpio_toggle(GPIOG, GPIO14);  // Hace que el LED parpadee
            for (int i = 0; i < LED_BLINK_INTERVAL; i++) {
                __asm__("nop");
            }
        } else {
            gpio_clear(GPIOG, GPIO14);  // Apaga el LED
        }

        sprintf(lcd_out, "Raw ADC: %d", raw_adc_value);
        gfx_setCursor(15, 198);
        gfx_puts(lcd_out);

        sprintf(lcd_out, "Voltaje: %.2fV", battery_voltage);
        gfx_setCursor(15, 250);
        gfx_puts(lcd_out);

        if (is_button_pressed()) {
        sprintf(usart_out, "%.2f\t%.2f\t%.2f\t%.2f\r\n", (float)x/100, (float)y
    /100, (float)z/100, battery_voltage);
        usart_send_string(usart_out);
```

```
        }

        lcd_show_frame();

        for (int i = 0; i < 8000; i++) {
            __asm__("nop");
        }
    }
    return 0;
}
```

Para la comunicación con thingsboard y la recepción de datos se implementó el siguiente script de Pyhton, en el broker se establece la página de thingsboard requerida y el token fue el generado al crear el dispositivo en thingsboard:

```
import paho.mqtt.client as mqtt
import serial
import json
import time

serial_port = serial.Serial(port="/dev/ttyACM0", baudrate=115200, timeout=1)

# Configuraci n MQTT para Thingsboard
broker = "iot.eie.ucr.ac.cr"
port = 1883
token = "e0823x0mia7dawj45ebl"

username = "STM32"


client = mqtt.Client()
client.username_pw_set(token)
client.connect(broker, port, 60)
client.loop_start()

def on_publish(client, userdata, result):
    print("data published to thingsboard")
    pass

client.on_publish = on_publish

while True:
    if serial_port.in_waiting > 0:
        line = serial_port.readline().decode('utf-8').strip()
        values = line.split("\t")  # Dividir los datos por tabulaciones

        # Suponiendo que tienes valores x, y, z y bater a en ese orden
        data = {
            "x": float(values[0]),
            "y": float(values[1]),
            "z": float(values[2]),
            "battery": float(values[3])
        }

        client.publish("v1/devices/me/telemetry", json.dumps(data), 1)
        time.sleep(1)

serial_port.close()
client.loop_stop()
```

Después de configurar esto y ejecutar el script de Pyhton se obtuvieron los siguientes resultados en el dashboard:

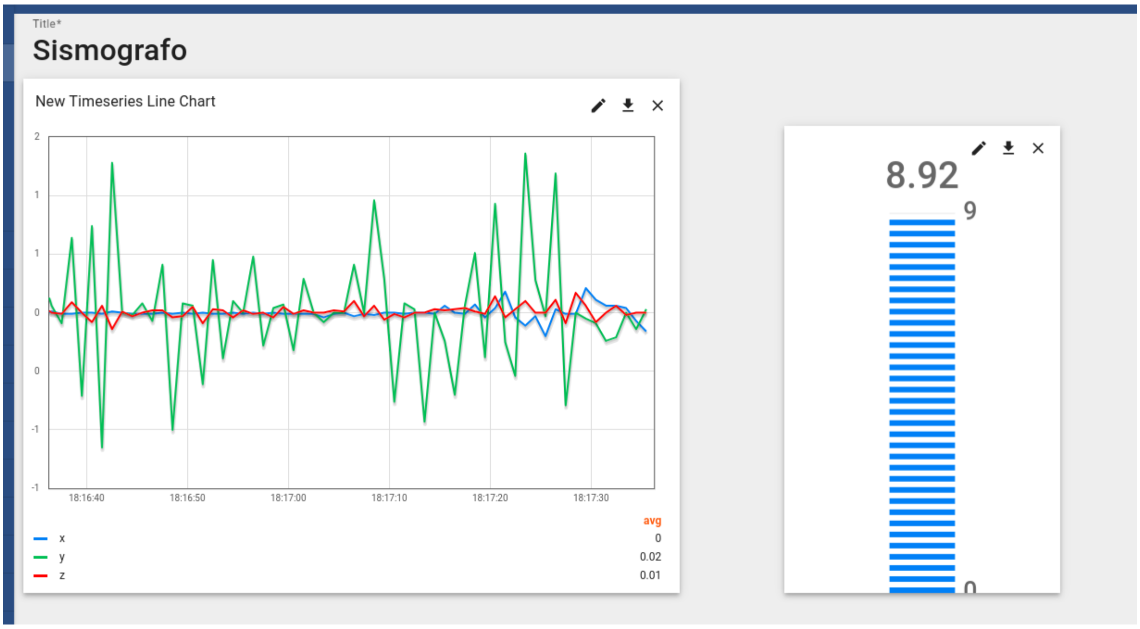


Figura 10: Resultados en Dashboard para batería y giroscopio

Finalmente se observa el manejo del led verde parpadeante cuando hay transmisión de datos al presionar el botón en la siguiente imagen:

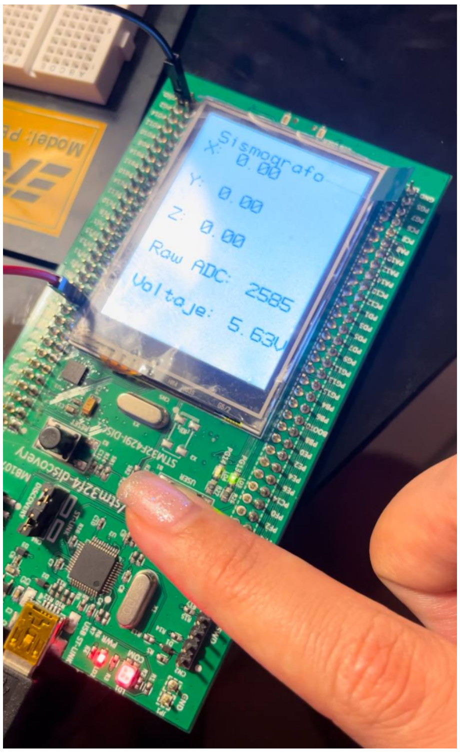


Figura 11: Parpadeo de led verde al iniciar la transmisión serial

También se adjuntará un video en la carpeta zip con el funcionamiento completo del circuito.

3.2. Repositorio Git

El repositorio de Github que contiene los archivos que muestran lo realizado se encuentra en el link: `https://github.com/SofiVillalta29/Laboratorio_De_Microcontroladores/tree/main`

4. Conclusiones y Recomendaciones

- Es muy importante para este laboratorio leer bien la hoja del fabricante para entender las tensiones aptas para el microcontrolador y además saber los pines disponibles para usar dependiendo de la aplicación, debido a que diferentes pines tienen diferentes funciones, por lo que puede haber un choque de señales, como por ejemplo con el giroscopio y la pantalla.
- Durante el desarrollo del laboratorio se destaca el uso de un divisor de tensión para controlar el ingreso de tensión al microcontrolador, al igual que un potenciómetro para simular la descarga dd ela batería, lo cual permitió evaluar el correcto funcionamiento dle circuito.
- Se concluye que el STM32F429i es un microcontrolador muy poderoso que se uede utilizar para diversas aplicaciones, y es sencillo utilizar la librería de libopem debido a que incluye varios ejemplos que pueden ser reutilizados para otros usos.

Referencias

[1] STMicroelectronics. 32b.Product reference manual. Disponible en: https://www.renesas.com/us/en/document/dst/ra4m3-group-datasheet

[2] STM32F429 Discovery kit. Product reference manual. Disponible en: `https://octopart.com/datasheet/stm32f429i-disco-stmicroelectronics-30261771`

[3] Componentes básicos en la electrónica. (s.f). Monografías.com. Disponible en: `https://www.monografias.com/trabajos107/componentes-basicos-electronica/componentes-basicos-electronica`

5. Apéndice

RA4M3 Group
Renesas Microcontrollers

R01DS0368EJ0140
Rev.1.40
Jul 28, 2023

Leading-performance 100 MHz Arm Cortex-M33 core, up to 1 MB code flash memory with background and SWAP operation, 8 KB Data flash memory, and 128 KB SRAM with Parity/ECC. High-integration with USB 2.0 Full-Speed, SDHI, Quad SPI, and advanced analog. Integrated Secure Crypto Engine with cryptography accelerators, key management support, tamper detection and power analysis resistance in concert with Arm TrustZone for integrated secure element functionality.

Features

■ Arm® Cortex®-M33 Core

- Armv8-M architecture with the main extension
- Maximum operating frequency: 100 MHz
- Arm Memory Protection Unit (Arm MPU)
  - Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU_S): 8 regions
  - Non-secure MPU (MPU_NS): 8 regions
- SysTick timer
  - Embeds two Systick timers: Secure and Non-secure instance
  - Driven by LOCO or system clock
- CoreSight™ ETM-M33

■ Memory

- Up to 1-MB code flash memory
- 8-KB data flash memory (100,000 program/erase (P/E) cycles)
- 128-KB SRAM

■ Connectivity

- Serial Communications Interface (SCI) × 6
  - Asynchronous interfaces
  - 8-bit clock synchronous interface
  - Smart card interface
  - Simple IIC
  - Simple SPI
  - Manchester coding (SCI3, SCI4)
- I^2C bus interface (IIC) × 2
- Serial Peripheral Interface (SPI)
- Quad Serial Peripheral Interface (QSPI)
- USB 2.0 Full-Speed Module (USBFS)
- Control Area Network module (CAN) × 2
- SD/MMC Host Interface (SDHI)
- Serial Sound Interface Enhanced (SSIE)

■ Analog

- 12-bit A/D Converter (ADC12) × 2
  - 5 Msps at interleaving
- 12-bit D/A Converter (DAC12) × 2
- Temperature Sensor (TSN)

■ Timers

- General PWM Timer 32-bit (GPT32) × 4
- General PWM Timer 16-bit (GPT16) × 4
- Low Power Asynchronous General Purpose Timer (AGT) × 6
- Watchdog Timer (WDT)
- Independent Watchdog Timer (IWDT)

■ Security and Encryption

- Secure Crypto Engine 9
  - Symmetric algorithms: AES
  - Asymmetric algorithms: RSA, ECC, and DSA
  - Hash-value generation: SHA224, SHA256, GHASH
  - 128-bit unique ID
- Arm® TrustZone®
  - Up to three regions for the code flash
  - Up to two regions for the data flash
  - Up to three regions for the SRAM
  - Individual secure or non-secure security attribution for each peripheral
- Device lifecyle management
- Pin function
  - Up to three tamper pins
  - Secure pin multiplexing

■ System and Power Management

- Low power modes
- Battery backup function (VBATT)
- Realtime Clock (RTC) with calendar and VBATT support
- Event Link Controller (ELC)
- Data Transfer Controller (DTC)
- DMA Controller (DMAC) × 8
- Power-on reset
- Low Voltage Detection (LVD) with voltage settings

■ Human Machine Interface (HMI)

- Capacitive Touch Sensing Unit (CTSU)

■ Multiple Clock Sources

- Main clock oscillator (MOSC) (8 to 24 MHz)
- Sub-clock oscillator (SOSC) (32.768 kHz)
- High-speed on-chip oscillator (HOCO) (16/18/20 MHz)
- Middle-speed on-chip oscillator (MOCO) (8 MHz)
- Low-speed on-chip oscillator (LOCO) (32.768 kHz)
- IWDT-dedicated on-chip oscillator (15 kHz)
- Clock trim function for HOCO/MOCO/LOCO
- PLL/PLL2
- Clock out support

■ General-Purpose I/O Ports

- 5-V tolerance, open drain, input pull-up, switchable driving ability

■ Operating Voltage

- VCC: 2.7 to 3.6 V

■ Operating Temperature and Packages

- Ta = -40°C to +105°C
  - 144-pin LQFP (20 mm × 20 mm, 0.5 mm pitch)
  - 100-pin LQFP (14 mm × 14 mm, 0.5 mm pitch)
  - 64-pin LQFP (10 mm × 10 mm, 0.5 mm pitch)
  - 144-pin BGA (7 mm × 7 mm, 0.50 mm pitch)
  - 64-pin BGA (6 mm × 6 mm, 0.65 mm pitch)

1. Overview

The MCU integrates multiple series of software- and pin-compatible Arm®-based 32-bit cores that share a common set of Renesas peripherals to facilitate design scalability and efficient platform-based product development.

The MCU in this series incorporates a high-performance Arm Cortex®-M33 core running up to 100 MHz with the following features:

- Up to 1 MB code flash memory
- 128 KB SRAM
- Quad Serial Peripheral Interface (QSPI)
- USBFS, SD/MMC Host Interface
- Capacitive Touch Sensing Unit (CTSU)
- Analog peripherals
- Security and safety features

1.1 Function Outline

Table 1.1 Arm core

Feature	Functional description
Arm Cortex-M33 core	● Maximum operating frequency: up to 100 MHz ● Arm Cortex-M33 core: – Armv8-M architecture with security extension – Revision: r0p4-00rel0 ● Arm Memory Protection Unit (Arm MPU) – Protected Memory System Architecture (PMSAv8) – Secure MPU (MPU_S): 8 regions – Non-secure MPU (MPU_NS): 8 regions ● SysTick timer – Embeds two Systick timers: Secure and Non-secure instance – Driven by SysTick timer clock (SYSTICCLK) or system clock (ICLK) ● CoreSight™ ETM-M33

Table 1.2 Memory

Feature	Functional description
Code flash memory	Maximum 1 MB of code flash memory.
Data flash memory	8 KB of data flash memory.
Option-setting memory	The option-setting memory determines the state of the MCU after a reset.
SRAM	On-chip high-speed SRAM with either parity bit or Error Correction Code (ECC).

Table 1.3 System (1 of 2)

Feature	Functional description
Operating modes	Two operating modes: ● Single-chip mode ● SCI/USB boot mode
Resets	The MCU provides 14 resets.
Low Voltage Detection (LVD)	The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The LVD module consists of three separate voltage level detectors (LVD0, LVD1, LVD2). LVD0, LVD1, and LVD2 measure the voltage level input to the VCC pin. LVD registers allow your application to configure detection of VCC changes at various voltage thresholds.

Table 1.3 System (2 of 2)

Feature	Functional description
Clocks	● Main clock oscillator (MOSC) ● Sub-clock oscillator (SOSC) ● High-speed on-chip oscillator (HOCO) ● Middle-speed on-chip oscillator (MOCO) ● Low-speed on-chip oscillator (LOCO) ● IWDT-dedicated on-chip oscillator ● PLL/PLL2 ● Clock out support
Clock Frequency Accuracy Measurement Circuit (CAC)	The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), the DMA Controller (DMAC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts.
Low power modes	Power consumption can be reduced in multiple ways, including setting clock dividers, stopping modules, selecting power control mode in normal operation, and transitioning to low power modes.
Battery backup function	A battery backup function is provided for partial powering by a battery. The battery-powered area includes the RTC, SOSC, backup memory, and switch between VCC and VBATT.
Register write protection	The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR).
Memory Protection Unit (MPU)	The MCU has one Memory Protection Unit (MPU).

Table 1.4 Event link

Feature	Functional description
Event Link Controller (ELC)	The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between the modules without CPU intervention.

Table 1.5 Direct memory access

Feature	Functional description
Data Transfer Controller (DTC)	A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request.
DMA Controller (DMAC)	The MCU includes an 8-channel direct memory access controller (DMAC) that can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

Table 1.6 External bus interface

Feature	Functional description
External bus	● QSPI area (EQBIU): Connected to the QSPI (external device interface)

Table 1.7 Timers (1 of 2)

Feature	Functional description
General PWM Timer (GPT)	The General PWM Timer (GPT) is a 32-bit timer with GPT32 × 4 channels and a 16-bit timer with GPT16 × 4 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer.
Port Output Enable for GPT (POEG)	The Port Output Enable (POEG) function can place the General PWM Timer (GPT) output pins in the output disable state

Table 1.7 Timers (2 of 2)

Feature	Functional description
Low Power Asynchronous General Purpose Timer (AGT)	The Low Power Asynchronous General Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register.
Realtime Clock (RTC)	For calendar count mode, the RTC has a 100-year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar.
Watchdog Timer (WDT)	The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt.
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers.

Table 1.8 Communication interfaces (1 of 2)

Feature	Functional description
Serial Communications Interface (SCI)	The Serial Communications Interface (SCI) × 6 channels have asynchronous and synchronous serial interfaces: ● Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA)) ● 8-bit clock synchronous interface ● Simple IIC (master-only) ● Simple SPI ● Smart card interface ● Manchester interface ● Extended Serial interface The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0, 3, 4, 9) has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator.
I^2C bus interface (IIC)	The I^2C bus interface (IIC) has 2 channels. The IIC module conforms with and provides a subset of the NXP I^2C (Inter-Integrated Circuit) bus interface functions.
Serial Peripheral Interface (SPI)	The Serial Peripheral Interface (SPI) has 1 channel. The SPI provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices.
Control Area Network (CAN)	The Controller Area Network (CAN) module uses a message-based protocol to receive and transmit data between multiple slaves and masters in electromagnetically noisy applications. The module complies with the ISO 11898-1 (CAN 2.0A/CAN 2.0B) standard and supports up to 32 mailboxes, which can be configured for transmission or reception in normal mailbox and FIFO modes. Both standard (11-bit) and extended (29-bit) messaging formats are supported. The CAN module requires an additional external CAN transceiver.
USB 2.0 Full-Speed module (USBFS)	The USB 2.0 Full-Speed module (USBFS) can operate as a host controller or device controller. The module supports full-speed and low-speed (host controller only) transfer as defined in Universal Serial Bus Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in Universal Serial Bus Specification 2.0. The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on the peripheral devices used for communication or based on your system.
Quad Serial Peripheral Interface (QSPI)	The Quad Serial Peripheral Interface (QSPI) is a memory controller for connecting a serial ROM (nonvolatile memory such as a serial flash memory, serial EEPROM, or serial FeRAM) that has an SPI-compatible interface.

Table 1.8 Communication interfaces (2 of 2)

Feature	Functional description
Serial Sound Interface Enhanced (SSIE)	The Serial Sound Interface Enhanced (SSIE) peripheral provides functionality to interface with digital audio devices for transmitting I^2S/Monaural/TDM audio data over a serial bus. The SSIE supports an audio clock frequency of up to 50 MHz, and can be operated as a slave or master receiver, transmitter, or transceiver to suit various applications. The SSIE includes 32-stage FIFO buffers in the receiver and transmitter, and supports interrupts and DMA-driven data reception and transmission.
SD/MMC Host Interface (SDHI)	The SDHI and MultiMediaCard (MMC) interface module provides the functionality required to connect a variety of external memory cards to the MCU. The SDHI supports both 1- and 4-bit buses for connecting memory cards that support SD, SDHC, and SDXC formats. When developing host devices that are compliant with the SD Specifications, you must comply with the SD Host/Ancillary Product License Agreement (SD HALA). The MMC interface supports 1-bit, 4-bit, and 8-bit MMC buses that provide eMMC 4.51 (JEDEC Standard JESD 84-B451) device access. This interface also provides backward compatibility and supports high-speed SDR transfer modes.

Table 1.9 Analog

Feature	Functional description
12-bit A/D Converter (ADC12)	Two units of 12-bit successive approximation A/D converter (ADC12) are provided. Analog input channels are selectable up to 12 in unit 0 and up to 10 in unit 1. Each 3 analog input of unit 0 and unit 1 is assigned to the same port (AN000/AN100, AN001/AN101, and AN002/AN102), and up to 19 ports are available as analog input. The temperature sensor output and an internal reference voltage are selectable for conversion in each unit 0 and unit 1.
12-bit D/A Converter (DAC12)	A 12-bit D/A converter (DAC12) is provided.
Temperature Sensor (TSN)	The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC12 for conversion and can be further used by the end application.

Table 1.10 Human machine interfaces

Feature	Functional description
Capacitive Touch Sensing Unit (CTSU)	The Capacitive Touch Sensing Unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software that enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with an electrical conductor so that a finger does not come into direct contact with the electrode.

Table 1.11 Data processing

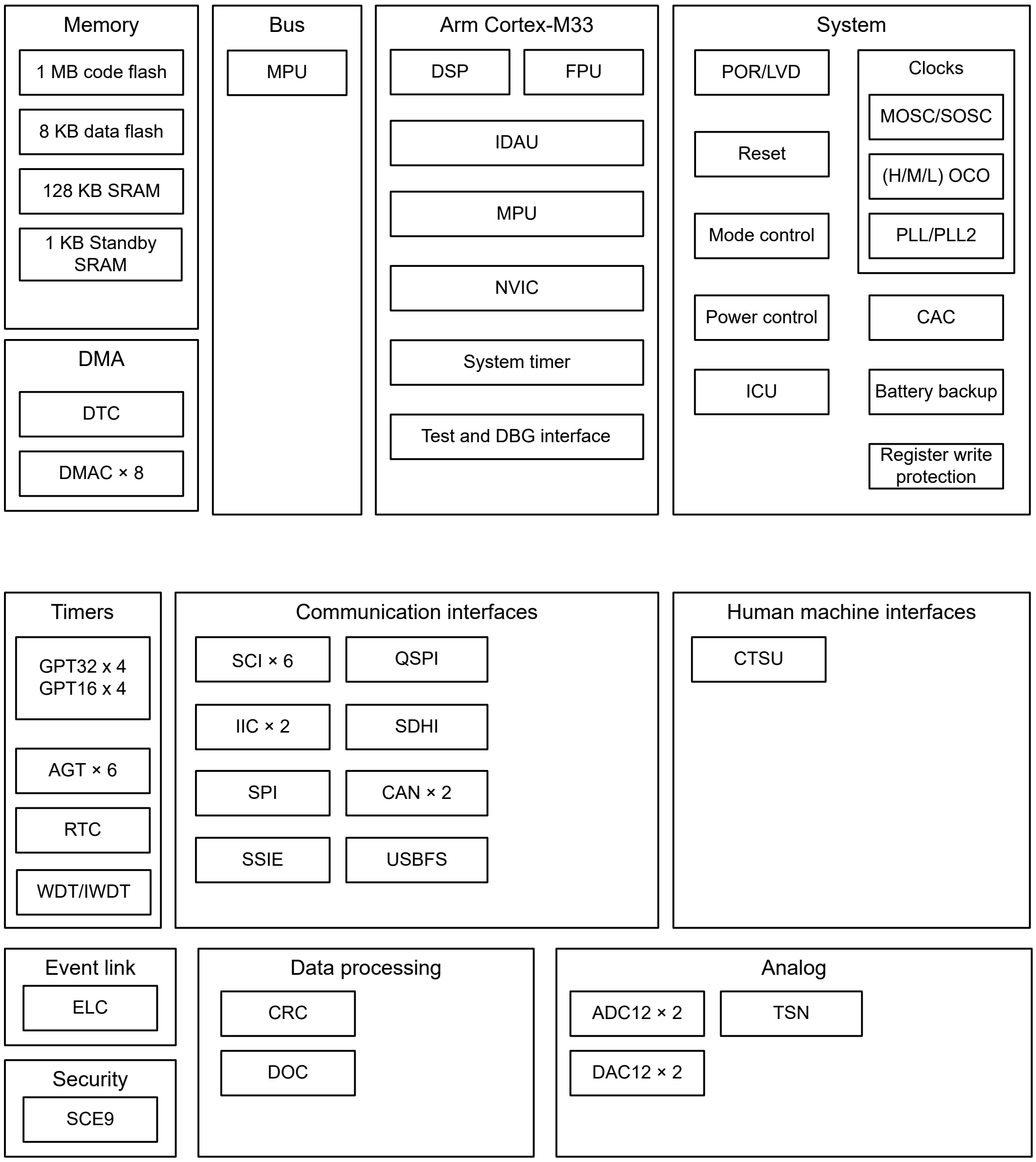
Feature	Functional description
Cyclic Redundancy Check (CRC) calculator	The Cyclic Redundancy Check (CRC) generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available.
Data Operation Circuit (DOC)	The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. When a selected condition applies, 16-bit data is compared and an interrupt can be generated.

Table 1.12 I/O ports

Feature	Functional description
Programmable I/O ports	● I/O ports for the 144-pin LQFP – I/O pins: 109 – Input pins: 1 – Pull-up resistors: 110 – N-ch open-drain outputs: 109 – 5-V tolerance: 21 ● I/O ports for the 144-pin BGA – I/O pins: 109 – Input pins: 1 – Pull-up resistors: 110 – N-ch open-drain outputs: 109 – 5-V tolerance: 21 ● I/O ports for the 100-pin LQFP – I/O pins: 75 – Input pins: 1 – Pull-up resistors: 76 – N-ch open-drain outputs: 75 – 5-V tolerance: 14 ● I/O ports for the 64-pin LQFP – I/O pins: 41 – Input pins: 1 – Pull-up resistors: 42 – N-ch open-drain outputs: 41 – 5-V tolerance: 9 ● I/O ports for the 64-pin BGA – I/O pins: 45 – Input pins: 1 – Pull-up resistors: 46 – N-ch open-drain outputs: 45 – 5-V tolerance: 9

1.2 Block Diagram

Figure 1.1 shows a block diagram of the MCU superset. Some individual devices within the group have a subset of the features.

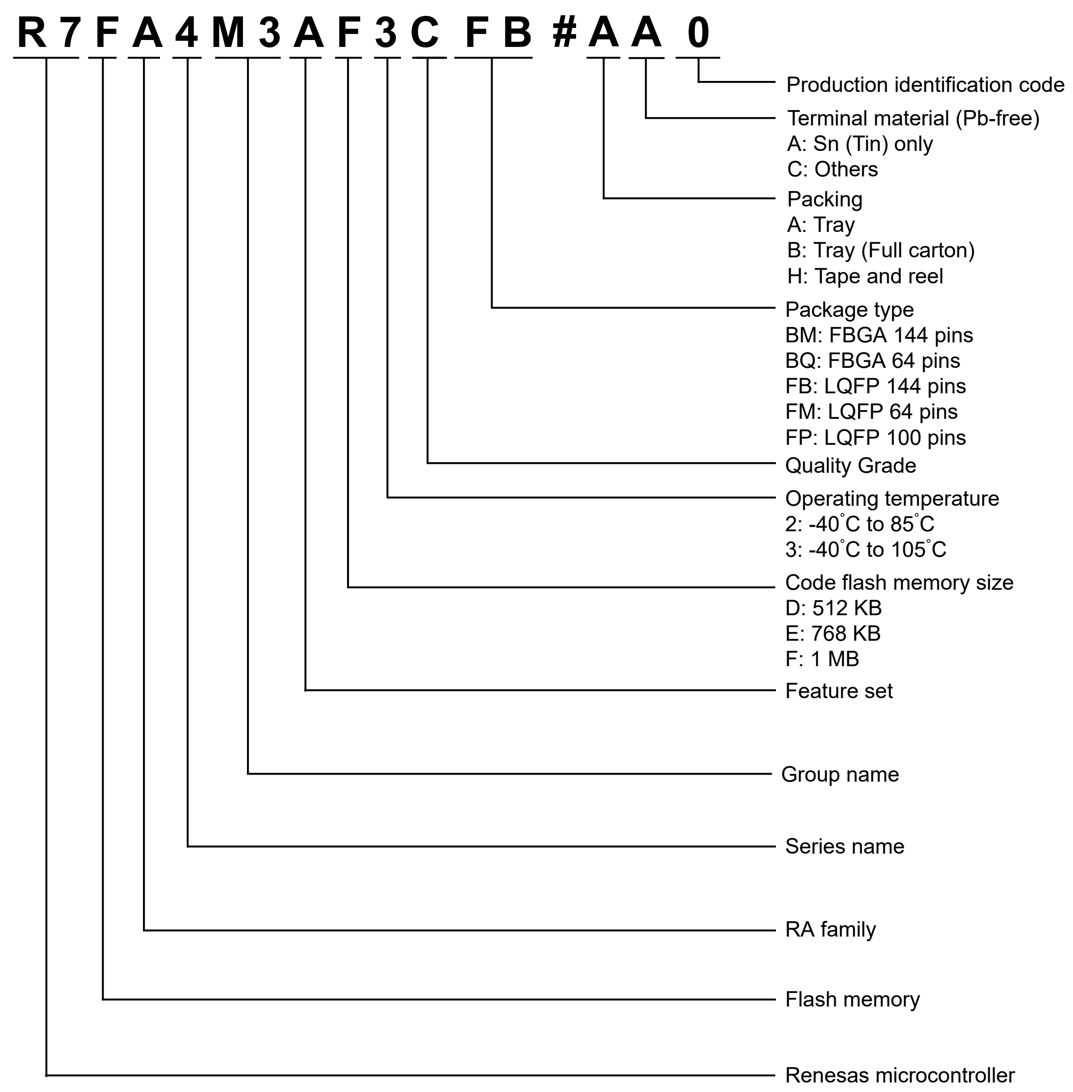


Note: Not available on all parts.

Figure 1.1 Block diagram

1.3 Part Numbering

Figure 1.2 shows the product part number information, including memory capacity and package type. Table 1.13 shows a list of products.



Note: Check the order screen for each product on the Renesas website for valid symbols after the #.

Figure 1.2 Part numbering scheme

Table 1.13 Product list (1 of 2)

Product part number	Package code	Code flash	Data flash	SRAM	Operating temperature
R7FA4M3AF2CBM	PLBG0144KB-A	1 MB	8 KB	128 KB	-40 to +85°C
R7FA4M3AF3CBM	PLBG0144KB-A				-40 to +105°C
R7FA4M3AF3CFB	PLQP0144KA-B				
R7FA4M3AF3CFP	PLQP0100KB-B				
R7FA4M3AF2CBQ	PLBG0064JC-A				-40 to +85°C
R7FA4M3AF3CBQ	PLBG0064JC-A				-40 to +105°C
R7FA4M3AF3CFM	PLQP0064KB-C				

Table 1.13 Product list (2 of 2)

Product part number	Package code	Code flash	Data flash	SRAM	Operating temperature
R7FA4M3AE2CBM	PLBG0144KB-A	768 KB	8 KB	128 KB	-40 to +85°C
R7FA4M3AE3CBM	PLBG0144KB-A				-40 to +105°C
R7FA4M3AE3CFB	PLQP0144KA-B				
R7FA4M3AE3CFP	PLQP0100KB-B				
R7FA4M3AE2CBQ	PLBG0064JC-A				-40 to +85°C
R7FA4M3AE3CBQ	PLBG0064JC-A				-40 to +105°C
R7FA4M3AE3CFM	PLQP0064KB-C				
R7FA4M3AD2CBM	PLBG0144KB-A	512 KB	8 KB	128 KB	-40 to +85°C
R7FA4M3AD3CBM	PLBG0144KB-A				-40 to +105°C
R7FA4M3AD3CFB	PLQP0144KA-B				
R7FA4M3AD2CBQ	PLBG0064JC-A				-40 to +85°C
R7FA4M3AD3CBQ	PLBG0064JC-A				-40 to +105°C

1.4 Function Comparison

Table 1.14 Function Comparison (1 of 2)

<table>
<tr><th colspan="2">Parts number</th><th>R7FA4M3AXXCFB</th><th>R7FA4M3AXXCBM</th><th>R7FA4M3AXXCFP</th><th>R7FA4M3AXXCFM</th><th>R7FA4M3AXXCBQ</th></tr>
<tr><td colspan="2">Pin count</td><td colspan="2">144</td><td>100</td><td colspan="2">64</td></tr>
<tr><td colspan="2">Package</td><td>LQFP</td><td>BGA</td><td colspan="2">LQFP</td><td>BGA</td></tr>
<tr><td colspan="2">Code flash memory</td><td colspan="2">1 MB
768 KB
512 KB</td><td colspan="2">1 MB
768 KB</td><td>1 MB
768 KB
512 KB</td></tr>
<tr><td colspan="2">Data flash memory</td><td colspan="5">8 KB</td></tr>
<tr><td colspan="2">SRAM</td><td colspan="5">128 KB</td></tr>
<tr><td></td><td>Parity</td><td colspan="5">64 KB</td></tr>
<tr><td></td><td>ECC</td><td colspan="5">64 KB</td></tr>
<tr><td colspan="2">Standby SRAM</td><td colspan="5">1 KB</td></tr>
<tr><td rowspan="2">DMA</td><td>DTC</td><td colspan="5">Yes</td></tr>
<tr><td>DMAC</td><td colspan="5">8</td></tr>
<tr><td rowspan="5">System</td><td>CPU clock</td><td colspan="5">100 MHz (max.)</td></tr>
<tr><td>CPU clock sources</td><td colspan="5">MOSC, SOSC, HOCO, MOCO, LOCO, PLL</td></tr>
<tr><td>CAC</td><td colspan="5">Yes</td></tr>
<tr><td>WDT/ IWDT</td><td colspan="5">Yes</td></tr>
<tr><td>Backup register</td><td colspan="5">128 B</td></tr>
<tr><td rowspan="8">Communication</td><td>SCI</td><td colspan="5">6</td></tr>
<tr><td>IIC</td><td colspan="5">2</td></tr>
<tr><td>SPI</td><td colspan="5">1</td></tr>
<tr><td>CAN</td><td colspan="5">2</td></tr>
<tr><td>USBFS</td><td colspan="5">Yes</td></tr>
<tr><td>QSPI</td><td colspan="5">Yes</td></tr>
<tr><td>SSIE</td><td colspan="3">Yes</td><td>No</td><td>Yes</td></tr>
<tr><td>SDHI/MMC</td><td colspan="3">Yes</td><td colspan="2">No</td></tr>
<tr><td rowspan="4">Timers</td><td>GPT32[*1]</td><td colspan="5">4</td></tr>
<tr><td>GPT16[*1]</td><td colspan="5">4</td></tr>
<tr><td>AGT[*1]</td><td colspan="5">6</td></tr>
<tr><td>RTC</td><td colspan="5">Yes</td></tr>
<tr><td rowspan="3">Analog</td><td>ADC12</td><td colspan="2">Unit 0: 12
Unit 1: 10
Shared channel pin: 3[*2]</td><td>Unit 0: 11
Unit 1: 9
Shared channel pin: 3[*2]</td><td colspan="2">Unit 0: 7
Unit 1: 4
Shared channel pin: 3[*2]</td></tr>
<tr><td>DAC12</td><td colspan="5">2</td></tr>
<tr><td>TSN</td><td colspan="5">Yes</td></tr>
<tr><td>HMI</td><td>CTSU</td><td colspan="2">20</td><td>12</td><td colspan="2">7</td></tr>
</table>

Table 1.14 Function Comparison (2 of 2)

Parts number		R7FA4M3AXXCFB	R7FA4M3AXXCBM	R7FA4M3AXXCFP	R7FA4M3AXXCFM	R7FA4M3AXXCBQ
Data processing	CRC	Yes				
	DOC	Yes				
Event control	ELC	Yes				
Security		SCE9, TrustZone, and Lifecycle management				
I/O ports	I/O pins	109	109	75	41	45
	Input pins	1	1	1	1	1
	Pull-up resistors	110	110	76	42	46
	N-ch open-drain outputs	109	109	75	41	45
	5-V tolerance	21	21	14	9	9

Note 1. Available pins depend on the Pin count, about details see section 1.7. Pin Lists.
Note 2. Some input channels of the ADC units are sharing same port pin.

1.5 Pin Functions

Table 1.15 Pin functions (1 of 4)

Function	Signal	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply. Connect this pin to VSS by a 0.1-μF capacitor. The capacitor should be placed close to the pin.
	VCL/VCL0	I/O	Connect this pin to the VSS pin by the smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VBATT	Input	Battery Backup power pin
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCOUT	Output	
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pin for setting the operating mode. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Measurement reference clock input pin
On-chip emulator	TMS	Input	On-chip emulator or boundary scan pins
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TCLK	Output	Output clock for synchronization with the trace data
	TDATA0 to TDATA3	Output	Trace data output
	SWO	Output	Serial wire trace output pin
	SWDIO	I/O	Serial wire debug data input/output pin
	SWCLK	Input	Serial wire clock pin
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQn	Input	Maskable interrupt request pins
	IRQn-DS	Input	Maskable interrupt request pins that can also be used in Deep Software Standby mode

Table 1.15 Pin functions (2 of 4)

Function	Signal	I/O	Description
GPT	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input pins
	GTIOCnA, GTIOCnB	I/O	Input capture, output compare, or PWM output pins
	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W phase)
AGT	AGTEEn	Input	External event input enable signals
	AGTIOn	I/O	External event input and pulse output pins
	AGTOn	Output	Pulse output pins
	AGTOAn	Output	Output compare match A output pins
	AGTOBn	Output	Output compare match B output pins
RTC	RTCOUT	Output	Output pin for 1-Hz or 64-Hz clock
	RTCICn	Input	Time capture event input pins
SCI	SCKn	I/O	Input/output pins for the clock (clock synchronous mode)
	RXDn	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXDn	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTSn_RTSn	I/O	Input/output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode), active-low.
	CTSn	Input	Input for the start of transmission.
	SCLn	I/O	Input/output pins for the IIC clock (simple IIC mode)
	SDAn	I/O	Input/output pins for the IIC data (simple IIC mode)
	SCKn	I/O	Input/output pins for the clock (simple SPI mode)
	MISOn	I/O	Input/output pins for slave transmission of data (simple SPI mode)
	MOSIn	I/O	Input/output pins for master transmission of data (simple SPI mode)
	RXDXn	Input	Input pins for received data (Extended Serial Mode)
	TXDXn	Output	Output pins for transmitted data (Extended Serial Mode)
	SIOXn	I/O	Input/output pins for received or transmitted data (Extended Serial Mode)
	SSn	Input	Chip-select input pins (simple SPI mode), active-low
IIC	SCLn	I/O	Input/output pins for the clock
	SDAn	I/O	Input/output pins for data

Table 1.15 Pin functions (3 of 4)

Function	Signal	I/O	Description
SPI	RSPCKA	I/O	Clock input/output pin
	MOSIA	I/O	Input or output pins for data output from the master
	MISOA	I/O	Input or output pins for data output from the slave
	SSLA0	I/O	Input or output pin for slave selection
	SSLA1 to SSLA3	Output	Output pins for slave selection
CAN	CRXn	Input	Receive data
	CTXn	Output	Transmit data
USBFS	VCC_USB	Input	Power supply pin
	VSS_USB	Input	Ground pin
	USB_DP	I/O	D+ pin of the USB on-chip transceiver. Connect this pin to the D+ pin of the USB bus.
	USB_DM	I/O	D- pin of the USB on-chip transceiver. Connect this pin to the D- pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. Connect this pin to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB_OVRCURA, USB_OVRCURB	Input	Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_OVRCURA-DS, USB_OVRCURB-DS	Input	Overcurrent pins for USBFS that can also be used in Deep Software Standby mode. Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_ID	Input	Connect the MicroAB connector ID input signal to this pin during operation in OTG mode
QSPI	QSPCLK	Output	QSPI clock output pin
	QSSL	Output	QSPI slave output pin
	QIO0 to QIO3	I/O	Data0 to Data3
SSIE	SSIBCK0	I/O	SSIE serial bit clock pins
	SSILRCK0/SSIFS0	I/O	LR clock/frame synchronization pins
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
	SSIDATA0	I/O	Serial data input/output pins
	AUDIO_CLK	Input	External clock pin for audio (input oversampling clock)
SDHI/MMC	SD0CLK	Output	SD/MMC clock output pins
	SD0CMD	I/O	Command output pin and response input signal pins
	SD0DAT0 to SD0DAT7	I/O	SD/MMC data bus pins
	SD0CD	Input	SD/MMC card detection pins
	SD0WP	Input	SD/MMC write-protect signals

Table 1.15 Pin functions (4 of 4)

Function	Signal	I/O	Description
Analog power supply	AVCC0	Input	Analog voltage supply pin. This is used as the analog power supply for the respective modules. Supply this pin with the same voltage as the VCC pin.
	AVSS0	Input	Analog ground pin. This is used as the analog ground for the respective modules. Supply this pin with the same voltage as the VSS pin.
	VREFH	Input	Analog reference voltage supply pin for the ADC12 (unit 1) and D/A Converter. Connect this pin to AVCC0 when not using the ADC12 (unit 1) and D/A Converter.
	VREFL	Input	Analog reference ground pin for the ADC12 and D/A Converter. Connect this pin to AVSS0 when not using the ADC12 (unit 1) and D/A Converter.
	VREFH0	Input	Analog reference voltage supply pin for the ADC12 (unit 0). Connect this pin to AVCC0 when not using the ADC12 (unit 0).
	VREFL0	Input	Analog reference ground pin for the ADC12. Connect this pin to AVSS0 when not using the ADC12 (unit 0).
ADC12	ANmn	Input	Input pins for the analog signals to be processed by the A/D converter. (m: ADC unit number, n: pin number)
	ADTRGm	Input	Input pins for the external trigger signals that start the A/D conversion, active-low.
DAC12	DAn	Output	Output pins for the analog signals processed by the D/A converter.
CTSU	TSn	Input	Capacitive touch detection pins (touch pins)
	TSCAP	I/O	Secondary power supply pin for the touch driver
I/O ports	Pmn	I/O	General-purpose input/output pins (m: port number, n: pin number)
	P200	Input	General-purpose input pin

1.6 Pin Assignments

The following figures show the pin assignments from the top view.

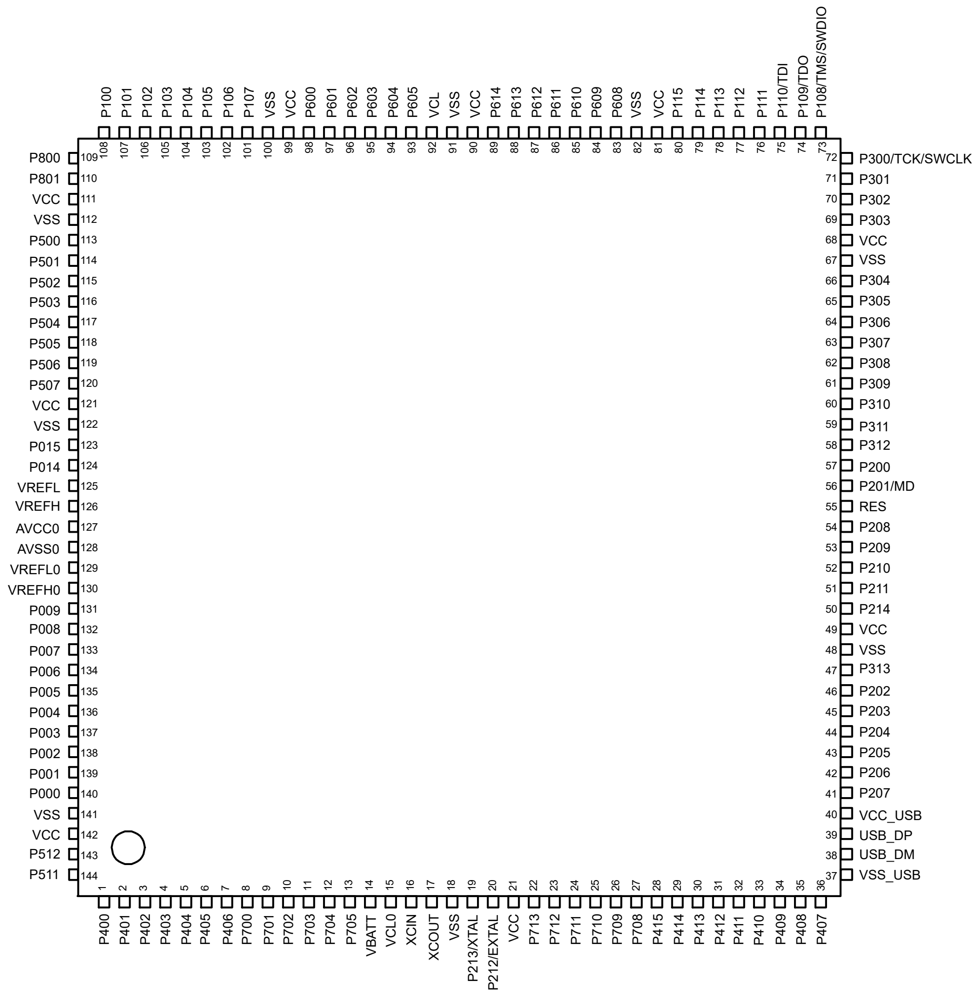


Figure 1.3 Pin assignment for LQFP 144-pin

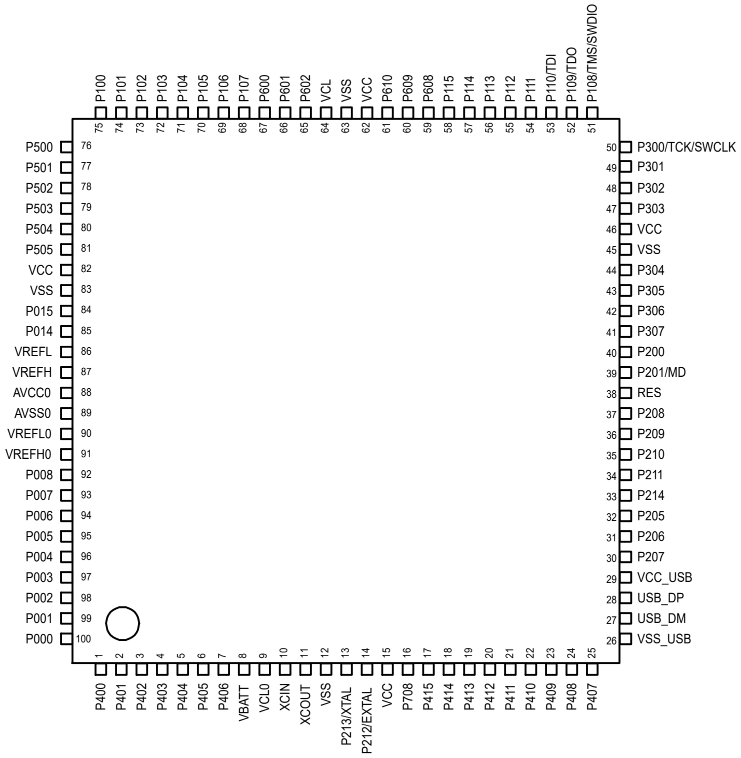


Figure 1.4 Pin assignment for LQFP 100-pin

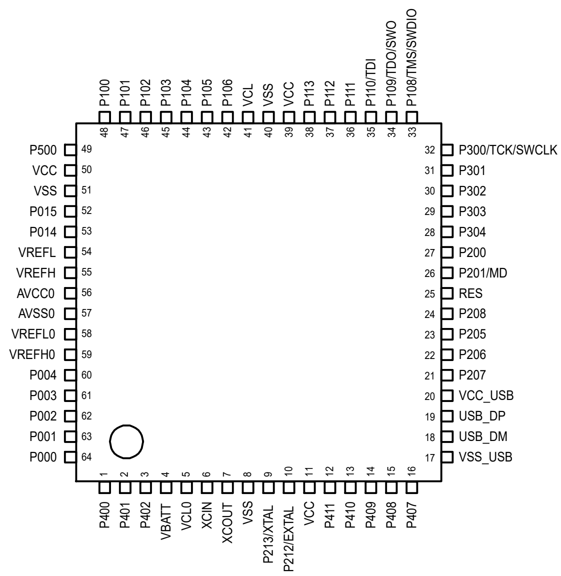


Figure 1.5 Pin assignment for LQFP 64-pin

	A	B	C	D	E	F	G	H	
8	P407	P212/ EXTAL	P213/ XTAL	XCOUT	XCIN	VCL0	VBATT	P000	8
7	USB_DM	P408	P409	P411	P405	P401	P002	P001	7
6	USB_DP	P207	P111	P410	P404	P400	P004	P003	6
5	P205	P206	P208	P113	P403	AVSS0	VREFL0	VREFH0	5
4	P200	P201/ MD	RES	P406	P402	AVCC0	VREFL	VREFH	4
3	P304	P303	P302	P106	P105	P104	P102	P014	3
2	P300/ TCK/ SWCLK	P301	P110/ TDI	VCC_USB	VSS_USB	P103	P101	P015	2
1	P108/ TMS/ SWDIO	P109/ TDO/ SWO	P112	VCC	VCL	VSS	P100	P500	1
	A	B	C	D	E	F	G	H	

Figure 1.6 Pin assignment for BGA 64-pin (top view, pad side down)

	A	B	C	D	E	F	G	H	J	K	L	M	N	
13	P408	P410	P412	P212 /EXTAL	P213 /XTAL	XCOUT	XCIN	P705	P700	P405	P403	P400	P401	13
12	USB_DM	P407	P413	P414	P708	P711	P713	P703	P702	P404	P402	P511	P512	12
11	USB_DP	P409	P411	P415	P710	P712	P709	P704	P701	P406	P001	P002	P000	11
10	P206	P207	P205	VCC_USB	VSS_USB	VCC	VSS	VCL0	VBATT	VCC	P005	P004	P003	10
9	P202	P204	P203	VSS						VSS	P007	P009	P006	9
8	P211	P214	P313	VCC						P008	AVSS0	VREFL0	VREFH0	8
7	P208	P210	P209	VSS						VSS	AVCC0	VREFL	VREFH	7
6	P200	P201/MD	RES	VCC						VCC	P503	P014	P015	6
5	P309	P311	P310	VSS						VSS	P505	P507	P506	5
4	P305	P306	P312	VSS	VCC	VCC	VSS	VCL	VCC	VCC	P502	P501	P504	4
3	P304	P307	P308	P110/TDI	P114	P608	P610	P605	P603	P105	P102	P500	P801	3
2	P303	P301	P108/TMS /SWDIO	P109/TDO	P112	P612	P614	P604	P601	P107	P104	P101	P800	2
1	P302	P300/TCK /SWCLK	P111	P113	P115	P609	P611	P613	P602	P600	P106	P103	P100	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	

Figure 1.7 Pin assignment for BGA 144-pin (top view, pad side down)

1.7 Pin Lists

Table 1.16 Pin list (1 of 4)

LQFP144	BGA144	LQFP100	LQFP64	BGA64	Power, System, Clock, Debug, CAC	I/O ports	Ex. Interrupt	SCI/IIC/SPI/CAN/USBFS/QSPI/SSIE/SDHI/MMC	GPT/AGT/RTC	ADC12/DAC12	CTSU
1	M13	1	1	F6	—	P400	IRQ0	SCK4/SCL0_A/AUDIO_CLK	GTIOC6A/AGTIO1	ADTRG1	—
2	N13	2	2	F7	—	P401	IRQ5-DS	CTS4_RTS4/SS4/SDA0_A/CTX0	GTETRGA/GTIOC6B	—	—
3	L12	3	3	E4	CACREF	P402	IRQ4-DS	CTS4/CRX0/AUDIO_CLK	AGTIO0/AGTIO1/AGTIO2/AGTIO3/RTCIC0	—	—
4	L13	4	—	E5	—	P403	IRQ14-DS	SSIBCK0_A	GTIOC3A/AGTIO0/AGTIO1/AGTIO2/AGTIO3/RTCIC1	—	—
5	K12	5	—	E6	—	P404	IRQ15-DS	SSILRCK0/SSIFS0_A	GTIOC3B/AGTIO0/AGTIO1/AGTIO2/AGTIO3/RTCIC2	—	—
6	K13	6	—	E7	—	P405	—	SSITXD0_A	GTIOC1A	—	—
7	K11	7	—	D4	—	P406	—	SSLA3_C/SSIRXD0_A	GTIOC1B/AGTO5	—	—
8	J13	—	—	—	—	P700	—	MISOA_C	GTIOC5A/AGTO4	—	—
9	J11	—	—	—	—	P701	—	MOSIA_C	GTIOC5B/AGTO3	—	—
10	J12	—	—	—	—	P702	—	RSPCKA_C	GTIOC6A/AGTO2	—	—
11	H12	—	—	—	—	P703	—	SSLA0_C	GTIOC6B/AGTO1	—	—
12	H11	—	—	—	—	P704	—	SSLA1_C/CTX0	AGTO0	—	—
13	H13	—	—	—	—	P705	—	CTS3/SSLA2_C/CRX0	AGTIO0	—	—
14	J10	8	4	G8	VBATT	—	—	—	—	—	—
15	H10	9	5	F8	VCL0	—	—	—	—	—	—
16	G13	10	6	E8	XCIN	—	—	—	—	—	—
17	F13	11	7	D8	XCOUT	—	—	—	—	—	—
18	G10	12	8	F1	VSS	—	—	—	—	—	—
19	E13	13	9	C8	XTAL	P213	IRQ2	TXD1/MOSI1/SDA1/TXDX1/SIOX1	GTETRGC/GTIOC0A/AGTEE2	ADTRG1	—
20	D13	14	10	B8	EXTAL	P212	IRQ3	RXD1/MISO1/SCL1/RXDX1	GTETRGD/GTIOC0B/AGTEE1	—	—
21	F10	15	11	D1	VCC	—	—	—	—	—	—
22	G12	—	—	—	—	P713	—	—	GTIOC2A/AGTOA0	—	TS17
23	F11	—	—	—	—	P712	—	—	GTIOC2B/AGTOB0	—	TS16
24	F12	—	—	—	—	P711	—	CTS1_RTS1/SS1	AGTEE0	—	TS15
25	E11	—	—	—	—	P710	—	SCK1	—	—	TS14
26	G11	—	—	—	—	P709	IRQ10	TXD1/MOSI1/SDA1/TXDX1/SIOX1	—	—	TS13
27	E12	16	—	—	CACREF	P708	IRQ11	RXD1/MISO1/SCL1/RXDX1/AUDIO_CLK	—	—	TS12
28	D11	17	—	—	—	P415	IRQ8	USB_VBUSEN/SD0CD	GTIOC0A/AGTIO4	—	TS11
29	D12	18	—	—	—	P414	IRQ9	CTS0/SD0WP	GTIOC0B/AGTIO5	—	TS10
30	C12	19	—	—	—	P413	—	CTS0_RTS0/SSL0/SD0CLK_A	GTOUUP/AGTEE3	—	TS09
31	C13	20	—	—	—	P412	—	SCK0/CTS3/SD0CMD_A	GTOULO/AGTEE1	—	TS08
32	C11	21	12	D7	—	P411	IRQ4	TXD0/MOSI0/SDA0/CTS3_RTS3/SS3/SD0DAT0_A	GTOVUP/AGTOA1	—	TS07
33	B13	22	13	D6	—	P410	IRQ5	RXD0/MISO0/SCL0/SCK3/SD0DAT1_A	GTOVLO/AGTOB1	—	TS06
34	B11	23	14	C7	—	P409	IRQ6	TXD3/MOSI3/SDA3/USB_EXICEN	GTOWUP/AGTOA2	—	TS05
35	A13	24	15	B7	—	P408	IRQ7	CTS4/RXD3/MISO3/SCL3/SCL0_B/USB_ID	GTOWLO/GTIOC6B/AGTOB2	—	TS04
36	B12	25	16	A8	—	P407	—	CTS4_RTS4/SS4/SDA0_B/SSLA3_A/USB_VBUS	GTIOC6A/AGTIO0/RTCOUT	ADTRG0	TS03
37	E10	26	17	E2	VSS_USB	—	—	—	—	—	—
38	A12	27	18	A7	USB_DM	—	—	—	—	—	—
39	A11	28	19	A6	USB_DP	—	—	—	—	—	—
40	D10	29	20	D2	VCC_USB	—	—	—	—	—	—
41	B10	30	21	B6	—	P207	—	TXD4/MOSI4/SDA4/SSLA2_A/QSSL	—	—	TSCAP
42	A10	31	22	B5	—	P206	IRQ0-DS	RXD4/MISO4/SCL4/CTS9/SDA1_B/SSLA1_A/USB_VBUSEN/SSIDATA0_C/SD0DAT2_A	GTIU	—	TS02
43	C10	32	23	A5	CLKOUT	P205	IRQ1-DS	TXD4/MOSI4/SDA4/CTS9_RTS9/SS9/SCL1_B/SSLA0_A/USB_OVRCURA-DS/SSILRCK0/SSIFS0_C/SD0DAT3_A	GTIV/GTIOC4A/AGTO1	—	TS01

Table 1.16 Pin list (2 of 4)

LPQFP144	BGA144	LPQFP100	LPQFP64	BGA64	Power, System, Clock, Debug, CAC	I/O ports	Ex. Interrupt	SCI/IIC/SPI/CAN/USBFS/QSPI/SSIE/ SDHI/MMC	GPT/AGT/RTC	ADC12/DAC12	CTSU
44	B9	—	—	—	CACREF	P204	—	SCK4/SCK9/RSPCKA_A/USB_OVRCURB-DS/ SSIBCK0_C/SD0DAT4_A	GTIW/GTIOC4B/AGTIO1	—	TS00
45	C9	—	—	—	—	P203	IRQ2-DS	CTS2_RTS2/SS2/TXD9/MOSI9/SDA9/MOSIA_A/ CTX0/SD0DAT5_A	GTIOC5A/AGTOA3	—	TS18
46	A9	—	—	—	—	P202	IRQ3-DS	SCK2/RXD9/MISO9/SCL9/MISOA_A/CRX0/ SD0DAT6_A	GTIOC5B/AGTOB3	—	TS19
47	C8	—	—	—	—	P313	—	SD0DAT7_A	—	—	—
48	D9	—	—	—	VSS	—	—	—	—	—	—
49	D8	—	—	—	VCC	—	—	—	—	—	—
50	B8	33	—	—	TCLK	P214	—	QSPCLK/SD0CLK_B	GTIU/AGTO5	—	—
51	A8	34	—	—	TDATA0	P211	—	QIO0/SD0CMD_B	GTIV/AGTOA5	—	—
52	B7	35	—	—	TDATA1	P210	—	QIO1/SD0CD	GTIW/AGTOB5	—	—
53	C7	36	—	—	TDATA2	P209	—	QIO2/SD0WP	GTOVUP/AGTEE5	—	—
54	A7	37	24	C5	TDATA3	P208	—	QIO3/SD0DAT0_B	GTOVLO	—	—
55	C6	38	25	C4	RES	—	—	—	—	—	—
56	B6	39	26	B4	MD	P201	—	—	—	—	—
57	A6	40	27	A4	—	P200	NMI	—	—	—	—
58	C4	—	—	—	—	P312	—	CTS3_RTS3/SS3	AGTOA1	—	—
59	B5	—	—	—	—	P311	—	SCK3	AGTOB1	—	—
60	C5	—	—	—	—	P310	—	TXD3/MOSI3/SDA3/QIO3	AGTEE1	—	—
61	A5	—	—	—	—	P309	—	RXD3/MISO3/SCL3/QIO2	AGTOA4	—	—
62	C3	—	—	—	—	P308	—	CTS3/QIO1	AGTOB4	—	—
63	B3	41	—	—	—	P307	—	QIO0	GTOUUP_D/AGTEE4	—	—
64	B4	42	—	—	—	P306	—	QSSL	GTOULO_D/AGTOA2	—	—
65	A4	43	—	—	—	P305	IRQ8	QSPCLK	GTOWUP/AGTOB2	—	—
66	A3	44	28	A3	—	P304	IRQ9	—	GTOWLO/GTIOC7A/AGTEE2	—	—
67	D7	45	—	—	VSS	—	—	—	—	—	—
68	D6	46	—	—	VCC	—	—	—	—	—	—
69	A2	47	29	B3	—	P303	—	CTS9	GTIOC7B	—	—
70	A1	48	30	C3	—	P302	IRQ5	TXD2/MOSI2/SDA2/TXDX2/SIOX2/SSLA3_B	GTOUUP/GTIOC4A	—	—
71	B2	49	31	B2	—	P301	IRQ6	RXD2/MISO2/SCL2/RXDX2/CTS9_RTS9/SS9/ SSLA2_B	GTOULO/GTIOC4B/AGTIO0	—	—
72	B1	50	32	A2	TCK/SWCLK	P300	—	SSLA1_B	GTOUUP/GTIOC0A	—	—
73	C2	51	33	A1	TMS/SWDIO	P108	—	CTS9_RTS9/SS9/SSLA0_B	GTOULO/GTIOC0B/AGTOA3	—	—
74	D2	52	34	B1	TDO/SWO/CLKOUT	P109	—	TXD9/MOSI9/SDA9/MOSIA_B/CTX1	GTOVUP/GTIOC1A/AGTOB3	—	—
75	D3	53	35	C2	TDI	P110	IRQ3	CTS2_RTS2/SS2/RXD9/MISO9/SCL9/MISOA_B/ CRX1	GTOVLO/GTIOC1B/AGTEE3	—	—
76	C1	54	36	C6	—	P111	IRQ4	SCK2/SCK9/RSPCKA_B	GTIOC3A/AGTOA5	—	—
77	E2	55	37	C1	—	P112	—	TXD2/MOSI2/SDA2/TXDX2/SIOX2/SCK1/ SSLA0_B/QSSL/SSIBCK0_B	GTIOC3B/AGTOB5	—	—
78	D1	56	38	D5	—	P113	—	RXD2/MISO2/SCL2/RXDX2/SSILRCK0/SSIFS0_B	GTIOC2A/AGTEE5	—	—
79	E3	57	—	—	—	P114	—	CTS9/SSIRXD0_B	GTIOC2B/AGTIO5	—	—
80	E1	58	—	—	—	P115	—	SSITXD0_B	GTIOC4A	—	—
81	E4	—	—	—	VCC	—	—	—	—	—	—
82	D5	—	—	—	VSS	—	—	—	—	—	—
83	F3	59	—	—	—	P608	—	—	GTIOC4B	—	—
84	F1	60	—	—	—	P609	—	CTX1	GTIOC5A/AGTO5	—	—
85	G3	61	—	—	—	P610	—	CRX1	GTIOC5B/AGTO4	—	—
86	G1	—	—	—	CACREF/CLKOUT	P611	—	—	AGTO3	—	—
87	F2	—	—	—	—	P612	—	—	AGTO2	—	—
88	H1	—	—	—	—	P613	—	—	AGTO1	—	—
89	G2	—	—	—	—	P614	—	—	AGTO0	—	—

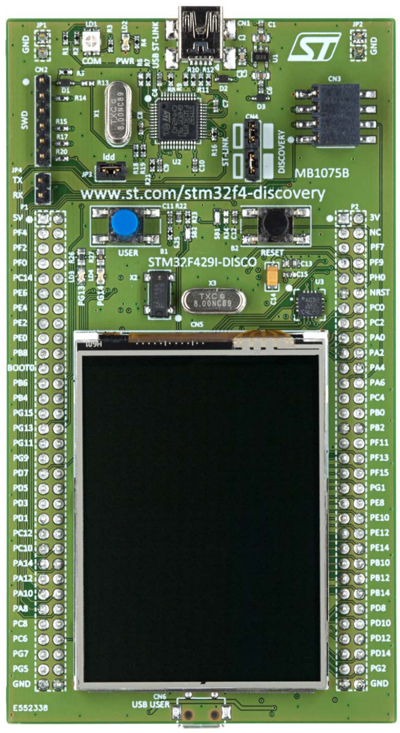
32F429IDISCOVERY

Discovery kit with STM32F429ZI MCU

Data brief

Features

- STM32F429ZIT6 microcontroller featuring 2 Mbytes of Flash memory, 256 Kbytes of RAM in an LQFP144 package
- On-board ST-LINK/V2 on STM32F429I-DISCO or ST-LINK/V2-B on STM32F429I-DISC1
- mbed™-enabled (mbed.org) with ST-LINK/V2-B only
- USB functions:
  - debug port
  - virtual COM port with ST-LINK/V2-B only
  - mass storage with ST-LINK/2-B only
- Board power supply: through the USB bus or from an external 3 V or 5 V supply voltage
- 2.4" QVGA TFT LCD
- 64-Mbit SDRAM
- L3GD20, ST MEMS motion sensor 3-axis digital output gyroscope
- Six LEDs:
  - LD1 (red/green) for USB communication
  - LD2 (red) for 3.3 V power-on
  - Two user LEDs: LD3 (green), LD4 (red)
  - Two USB OTG LEDs: LD5 (green) VBUS and LD6 (red) OC (over-current)
- Two push-buttons (user and reset)
- USB OTG with micro-AB connector
- Extension header for LQFP144 I/Os for a quick connection to the prototyping board and an easy probing
- Comprehensive free software including a variety of examples, part of STM32CubeF4 package or STSW-STM32138 for legacy standard libraries usage



1. Picture not contractual.

Description

The STM32F429 Discovery kit (32F429IDISCOVERY) allows users to easily develop applications with the STM32F429 high-performance MCUs with ARM® Cortex®-M4 core.

It includes an ST-LINK/V2 or ST-LINK/V2-B embedded debug tool, a 2.4" QVGA TFT LCD, an external 64-Mbit SDRAM, an ST MEMS gyroscope, a USB OTG micro-AB connector, LEDs and push-buttons.

System requirements

- Windows® OS (XP, 7, 8)
- USB type A to Mini-B cable

Development toolchains

- IAR EWARM (IAR Embedded Workbench®)
- Keil® MDK-ARM™
- GCC-based IDEs (free AC6: SW4STM32, Atollic® TrueSTUDIO®,...)
- ARM® mbed™ online

Demonstration software

The demonstration software is preloaded in the board Flash memory. It displays on the screen icons to run different applications: clock/calendar, a game, a video player and an image browser, performance monitoring and system information.

The latest versions of the demonstration source code and associated documentation can be downloaded from the www.st.com/stm32f4-discovery webpage.

Ordering information

To order the Discovery kit for the STM32F429 line of microcontrollers, refer to *Table 1*.

Table 1. List of the order codes

Order code	ST-LINK version
STM32F429I-DISCO	ST-LINK/V2
STM32F429I-DISC1	ST-LINK/V2-B (mbed-enabled)

Revision history

Table 2. Document revision history

Date	Revision	Changes
06-Sep-2013	1	Initial version.
29-Sep-2014	2	Updated *Section : Features* and *Section : Description* to introduce STM32cubeF4 and STSW-STM32138. Updated ST MEMS feature. Updated *Section : System requirements* and *Section : Development toolchains*.
23-Oct-2015	3	Updated *Section : Features*, *Section : Description*, *Section : Ordering information*.